

CSC2021 Advanced Programming using Android and C++  
CSC2022 Games Programming using Android and C++

# Student Advice

Words of wisdom from students  
who previously completed the  
project



## Advanced Programming

Philip Hanna CSC2021



## Games Programming

Philip Hanna CSC2022

At the end of the module students were asked if they would like to pass on some helpful advice to next year's students. This document contains their collective wisdom.

# Student Advice

All comments are reported as received. In some cases I have provided an aside to the comment. Remember, these comments are from students who have gone through what you will be going through. They really do offer words of wisdom.

The project has changed over the year (from individual to groups, from Java to C#/XNA to Android for this year). Most comments will still apply (e.g. learning Android will be much the same as learning C#). The comments are presented in chronological order (the most recent at the start).

Start as early as you can and learn the basics of C# and XNA.

Don't rely on forums! Build your knowledge as you go along.

Try to communicate with your team as much as possible and give each other different tasks to work on.

Don't let one team member do all the work... some people like to 'get things done' – make sure you don't get left behind by committing early.

Program early, program often. Don't panic. Take advantage of online resources in particular youtube.

Do a lot of core work early, free up the rest of the semester. Get a team you work well with.

Carefully choose your team members.

I really enjoyed working as part of a team. I would also say start early + keep upto date with what your other team members are doing.

If a member of your team is not pulling his own weight give him a warning. If he does not shape up then inform the lecturer. No point in carrying dead weight.

*Everyone had to work as part of team in the 2013/12 and 2012/2011 delivery. This meant a lot more team related advice was received – although, this quote is probably the most strongly worded one that was received!*

*Compulsory teams were introduced into the module given feedback from employers – virtually everyone works as part of a team within the computing sector (an aspect that most students will encounter as part of their placement year).*

*Getting a group of people to work effectively as a team is actually a rather difficult thing to manage (unless you happen to be lucky with your team composition). A good team will need to assign and manage workloads, tackling the problems (often unexpected) that will occur along the way. Clear communication alongside understanding and flexible team members are key to this.*

*If you find things are not working out for you within your team then please get in contact.*



Start early, everyone says that they will be most people won't, so honestly start early. Plan ahead, split your project into smaller areas and work around it first rather than trying to do it all at once.

Start earlier. Regular meetings.

Try and pick areas to work on with your team. Otherwise you may find you trip over each other when working separately.

Make sure you start the project early and try to get a basic version of the game running as quickly as possible. That will allow you to build on the basic version rather than panicking at the end as you try to get your game to function.

Try working closely with your team members, two heads working on the same code speeds along problem solving.

Work AS A TEAM – keep co-ordinated

Work towards an early 'base' state of the game which contains core features. Start early.

Distribute tasks to each team member from the very start. Meet at least twice a week. Get help with issues as soon as they occur, don't sit around.

Start looking at examples and tutorials for C# and XNA early on. Try to have a list of tasks that need done and agree on who'll be doing what.

Make sure you know your team! Dream big as you may not get everything you want in the game. Make sure to follow Phil as a medic in TF2 for instant success 😊

*I do love a bit of TF2 – and, admittedly, started playing the original TF back in my PhD years (I can still give you youngsters a run for your money). Although, healing won't equate to additional module marks.*



My advice is don't go in a team you're not completely happy with, also like everyone says, start early and maintain the work over the 12 weeks.

Start early, work together as a team. If things aren't going as expected as Phil. Think up wonderful, creative, hilarious questions for Phil to answer.

*Whilst I said on the last page that it was difficult for a group of people to work effectively within a team – the aside advice, if followed, will bring success to the vast majority of teams!*



This advice will be ignored much as I largely ignored it but DO NOT leave things to near the deadline. You'll regret it.

Start early and make sure to have regular team meetings and keep up to date with each others work.

Be aware if you've reached a deadlock. In every project I've undertaken, more or less, at some point we have reached a time where nobody's doing anything; there is always something to do. It's much more important to work consistently and constantly than to work quickly in bursts.

Put code into classes as soon as possible, hard to move out of main.

Start at week one or time will run away from you. Don't be too ambitious. Definitely try to be in a team with people you can work well with.

Try out tutorials online relevant to your game idea.

Make sure everyone works from the very start otherwise you will have issues.

Understand code written by teammates.

Do start as early as you can ... don't leave it to the last minute!!

Start as early as possible, get the textbook and start learning the language ASAP. Get yourself familiar with the software and its capabilities. Don't be afraid to ask Phil for help/advice and attend his problem solving sessions.

Start early and work through the exercises in the textbook.

Begin sooner...

If working as a team don't be afraid to ask for help.

Start early (earlier than I did at least!), even if that includes working through recommended text book, which you should get!

If you start off well keep up the momentum! Start looking at tutorials early on to build your understanding.

Chocolate chip & blueberry muffins are a good choice for Phil's muffin basket.

*The cohort 2013/14 and 2012/13 cohorts both hinted and suggested, on several occasions, that a muffin basket would be forthcoming. No muffin basket ever appeared.*

Google is your bible.  
Phil is your god.

*As a deity, I suggest weekly offerings of muffins.*



Manage time better or start earlier is said all of the time, but you don't realise until work piles up and time is lost.

Working as part of a team was good because shared ideas helped us to come together well and to create better solutions.

Spend time doing tutorials before jumping into your game.

Remember it will take a while to understand C# and XNA so take time to get your head around it.

Everyone will say to 'start your project early'. I would suggest that for the first few weeks (1-4) play about with online examples and get a feel for the code that you may be using, and to learn the extent of the features that you could put into your game. Do hard stuff first – means the last few weeks are free to fix bugs / add additional features. Don't be afraid to ask Phil for help!!

Choose a game idea which every member is enthusiastic about and wants to work on.

Start early, can't stress this enough. Our team start very early and even we are starting to get tight for time. Try to refine your game to the basic core and build it first, if you have more time then build upon

If you do this by yourself either ensure you already know C#/Java or get cracking from day 1! If you're doing this in a team, don't allow your friend's success to feel like your own -> some people stop working when one team member's working well.

*The later comment about team work is very valid. As we will be using versioning tools, it will be very easy to see who has done what at the end of the project. This plays an important factor within the assessment.*

The ability to bounce ideas off teammates allows you to be creative. In my opinion it is very important to find someone that this is possible with.

Start as early as possible so coming to the end you don't have the pressure and stress of rushing to get the game finished.

Make sure you stick with a team you know will work.

*The vast majority of projects were team based in 2011/12. Whilst this resulted in a rather good overall examination performance (failure rates were very low) it also meant that some people had to deal with the forms of team issue that arise now and again.*

*A team that experiences friction in terms of working well together will definitely offer you a valuable learning experience, albeit one that may not always be enjoyable.*

*Thankfully the assessment and marking process is flexible enough to take into account the range of team 'hic-ups' that might occur. If you are in anyway concerned with how your team is getting along then please do get in contact with me and we can try to resolve the problems.*



If you read the XNA textbook (up to page 150) it will provide the knowledge to tackle the project. Start early, try to find tutorials online. The more your program the more enjoyable and easier it becomes.

Start early to get up to speed on XNA as soon as possible to be able to start programming early in the semester. Ask Phil if you have any problem as he offers great advice.

Work consistently throughout the year, even if it is just a few hours a week and you'll be grand. Leave it til week 8 to start working in earnest (like I did) and you WILL regret it.



Beneficial to learn a bit of XNA before trying to develop the task as it can otherwise seem overwhelming.

Grasp the concepts of the new language faster than I did to allow you more time to explore their applications. In group work, establish a good communication between members and try to set and meet goals on time.



Break the task down into classes and outline contents on paper during design.

Try to start as early as you can, even if it is for a small amount of work. Use tutorials to improve understanding. Get in contact with Phil if you get stuck.

I did start quiet early but its always better to start from week 1 or 2 – especially given that week 10, 11 and 12 are the most busy weeks in the whole semester

*Please don't underestimate this. Most modules will have a large assignment, project, etc. due in around week 11 or 12.*

Try to tackle concepts yourself rather than relying completely on tutorials, you'll learn more.

Get stuck in from the very start. Make sure you talk to Phil about your problems as he has great advice on how you can make your game better. Why make things more difficult!

Plan ahead from the start and retain any notes from group sessions.

Start as early as possible and try to thoroughly plan your game

Have weekly meetings, goal setting, reviewing and define expectations.

Start early and do a small amount regularly

The flexibility of this course will allow you to do anything you want to explore, take advantage of this!

Regular meetings with team members both in and out of the lab helped bring together game ideas, solutions, etc. Regular commits to SVN in case something goes wrong. Spend a good deal of time planning the game and playing with code before starting the project.

Look at plenty of examples of code and try to find the resources you will need (images, etc.) before being fully set in on project design.

Pick your team carefully. Draw up a team contact/agreement before you start. Experiment early.

Have frequent meetings, with clear objectives and time-frames for completing them. Don't spend too long on the design before beginning as the design will evolve much more easily when you get stuck into coding!

Start at day one – plan, plan, plan. Make full use of Phil's online lectures (they're brill!). If in a team – make sure communication takes place regularly. Be prepared for team members to cross over on other team members tasks.

Spend time doing tutorials before jumping into your game.

Really structure objects and classes on paper and how they will communication between each other.

We planned the game well. However you need to start the game as early as possible and do not cram.

Keep an eye on group members work & relate closely to what there doing. Start early.

Start early. Don't slack. Design properly. Use all of the resources Master Phil provides. Set up XNA and VS early and start C# tutorials. Look at source code and buy the O'Reilly book.

My main advice would be to start as soon as you can, the earlier you start the more comfortable you will be as the weeks go on, don't panic and try do as much research as you can.

Sounds obvious... but start early. Try to have your game concept finalised ASAP. Ensure that you choose to work with people who have the same goals, work ethic, etc.

Start early, but then you already know that.

Sort out a clear idea of what exactly you want your game to do early on in the year. Do lots of testing.



# start!

Try to start doing the project as early as possible. Don't leave it until the last min. Check what resources are available (graphics) before you plan your game.

Weekly meetings with weekly targets!

Would it have been possible to do my game in sync with Phil's lectures, so that what was being discussed I could implement that week? I wonder this now. I wish I'd tried it. Also if you're reading this to procrastinate, go work already.

Start early and get stuck into it. Don't leave it to last few weeks.

Purchase Learning XNA 3.0 and read up-to and including Chapter 7 complete ALL exercises at the end of each chapter and experiment with the code yourself.



This module is a great darkness that will consume your university life. Beware!! .. also start early.

*Is this a bad thing? Surely not! Maybe I'm just a bad person.*

Start EARLY!! Really as soon as you get information on what to do start! Even if it's only getting images it will help later on.

Start thinking of what exactly you want to do and try not to make your game too complicated

START EARLY!! Also get a couple of good textbooks

Planning is of utmost importance.

Start early and work consistently throughout the year.

I start early and it definitely paid off towards the end.

If you found programming challenging in Level 1 you should seriously consider leaving the course.

Spend more time on the planning/design of the game architecture than you think you will need to.

*You will find a good few comments this year on the importance of planning the project (arguably the calls to start early as just another way of saying 'plan the project; well'). Ideally you should set yourself weekly targets- and make sure that you 'hit' your weekly deliverables!*

For XNA online tutorials are great for learning; but design and wiki your own code!

Everyone will say this, but start early. Get through at least the first 4 chapters in the textbook and their practical aspects so that you can get started with your code well in advance of the week 6 deadline.

Everyone will say this. Start as soon as you read this! It's not an especially hard project, but it is time consuming. Make life easy for yourself so you ain't stressed about your game, finding placement, databases or comp theory tutorials every week! Good luck!

*This module does build upon the material taught in Level 1 – importantly, the module also puts you in a position where you will have to demonstrate independent learning and independent problem solving (all core real-world programming skills). For most students this is something doable, for a small number of students this will be a real struggle. Whilst the failure rate on this module is not high, those who do fail do so either because things are left to the last minute or not enough time is put into the project. If you find progress is difficult then please don't ignore the problem until it's too late, instead talk to me and we can recast your project towards one that is more doable.*





Start early – time flies – best way to learn is practice.  
Don't be afraid to ask for help.

*Sage advice. The alpha hand-in in Week 9 is intended to provide a target point for an 'almost' completed game. Do plan to hit this target. Should you plan to hit this target you probably won't (!), but, and it's an important but, you'll likely get close and sufficiently close that your final Week 11 hand-in is stress free (relatively speaking).*

Start early. Set out as much functionality as possible as this is where 60 marks are allocated. Use Phil as much as possible as he is a great help. Get an early understanding of tutorials and XNA.

Definitely start early, pretend it has to be in earlier than it's supposed to be.

My advice would be to ask questions, don't be afraid not to know the answer, you were good enough to get to this stage of your studies so your skills are there. Take advantage of the lectures and library resources to get the answers you need.

START EARLY. I mean it. Time is going to go so fast....

Don't get obsessed with any new video games that are released when you are working on your project. Seriously. Spend time making games, not playing them!

You aren't going to be expected to know how to make a game in XNA from the get go. Learning it yourself is not as bad as you may expect. XNA is a much nicer program to use, and the XNA book is very helpful.

Start early. If someone is not producing something don't wait for them to do something, either kick them into gear or carry on regardless. Don't waste time.

Keep your game do simple that it seems stupid to you.

*There is real wisdom in this comment – it is much better to take a stupidly simple game idea and then build on it instead of a grand idea which will not be realised within the available time.*

My advice would be to ask questions, don't be afraid not to know the answer, you were good enough to get to this stage of your studies so your skills are there. Take advantage of the lectures and library resources to get the answers you need.

START EARLY. I mean it. Time is going to go so fast....

Don't get obsessed with any new video games that are released when you are working on your project. Seriously. Spend time making games, not playing them!

Don't panic if you have never worked with C#! I was very tempted to use Java for my project as the idea of learning another language scared me to death! But I'm glad I chose XNA; it's a great language for creating games, and there are plenty of tutorials to help you!



Start early in the year and do the 'to-do-lists' at the end of each lecture straight after the lecture!!

At the start do as many examples and practices as you can. Read books, tutorials, and samples; immerse yourself in XNA! But don't get complacent; manage your time, set internal milestones and don't leave it all to the end.

Start as soon as you can - don't leave it to the last minute. Make use of the XNA forums.

*A number of comments refer to the use of Internet resources, be this forums, tutorials, etc. I would very strongly encourage that you do so. Certainly I will be available to help and assist (although towards the end of the module I get swamped) although it would be unwise not to avail of the multitude of other sources beyond the module.*

Start right away, most modules have some form of assignment and time gets used up quickly. This is the biggest assignment but also has the most time to do it, so use it

Start early and work at your game every day - no joking....

Beware that not everyone within a team works equally. Pick a team carefully.

Using internet forums, tutorials, examples and existing code is very helpful. Taking apart and looking at other peoples code and coding techniques will help alot.

Don't panic if you have never worked with C#! I was very tempted to use Java for my project as the idea of learning another language scared me to death! But I'm glad I chose XNA; it's a great language for creating games, and there are plenty of tutorials to help you!

Gets early - get all the help you need from Phil he is really helpful

Try to attempt bits of code for your game, even if you think it won't work it means you can get your foot in the door. Plus, you can always change it.

Work steadily from the beginning, seriously. Do that and it's easy and fun. Have a good group, with people who know what they're doing/whose skills integrate well with yours.

Don't let advanced tutorials scare you, everything takes times to understand.

Think you have plenty of time to write your game? WRONG! Get started

Start early and don't lose momentum - cannot be stressed enough

Start straight away

My advice is to start early. Don't spend ages on the design, get stuck into the coding as soon as you can.

Start coding really early, never assume you are safe to take a break from it. You never are! Also games students, start other projects as soon as you get them so the last few weeks aren't totally crazy.

Start early. I started late and now I feel that I should have started the moment the project was given. Do not worry about XNA. It's very similar to Java

Everyone says start early, they are right. I did + still with one week to go have more improvements than I would like to have to do.

I can't stress enough spend your time in the early stages of the module to prepare and then prepare some more. It is very important for your success in later stages to plan. Work on tutorials, be confident to express yourself and enjoy yourself for making a game after all.

Start early if you are stuck on a particular problem, leave it and move on. Do not spend weeks trying to solve one problem. Find Phil Hanna's office and keep calling around to see him if you are stuck or completely confused. Pick a simple game idea and stick with it, if you are not a brilliant programmer from 1st year, this is the only way to go

Use XNA/C# it is good to learn and actually more capable! Don't be afraid to add constraints or drop features as you hit problems with it. The initial scale of the game you plan will probably hit problems – don't worry. By the end of the project you will probably have learned enough to wish you had taken a different direction possibly even have ideas for a completely different game. Don't worry, or try to change it. There's always your next game ☺

*This is very useful advice – the process of getting started on a programming project is, for most people, the most difficult bit. Once you get stuck into the coding things tend to get easier!*

## ADVICE

the right steps  
to take

Plan and start working on the game straight away, it takes a lot of time! Balance work between other modules. Use advisory sessions for more help.

Week 6 is NOT a starting point, use the early weeks to try tutorials and gain a greater knowledge of your language in the context of your game. You will be snowed under with work in the final weeks (especially large assessed practicals) do yourself a favour and start early, I haven't slept in 2 days... it's not even the final week yet.

Always begin early. 12 weeks seems like a long time but it's not really! Be organised and well planned

I read students' advice and they all said "start early!". It's true! You really need to get things moving as soon as you can. Don't put things off / don't think "we're making good progress, we can take it easy for a week". Keep working on the game as much as possible. It is a bit of a daunting task but you'll get through it. Phil cares about how your getting on & he genuinely wants you to do your best. Work hard and enjoy the relief when you hand it in! ☺

Start planning IMMEDIATELY! Time will disappear far too quickly!

You will hopefully spend a lot of time on the project so choose a game you will enjoy making so the module isn't a chore. I found it to be the most enjoyable module.

Work throughout the semester where help & advice is readily available. Do not leave most of the work to the last few weeks.

Actually plan your project- think about the structure before you begin. This saves a lot of time refactoring. And you should have started yesterday.

Plan! How will objects interact, what objects are needed, etc. having to refactor your code or just add hacks and workarounds is both time consuming, boring and stressful to debug. Start early but don't rush into coding. Design is so important! Understand your project before coding, but read sample code while your outlining your project. Ask as much as you can in the handins, the audio feedback is more helpful than you expect in terms of giving motivation and direction.

Everyone always says to start early. There's a good reason for that! Do as many tutorials as you can in the first few weeks to gain a better understanding of the things XNA or Java offers.

I strongly advise using C#. Not only because I like it better but because it's fairly easy to pick up and after a short period, you can become fairly creative with it. Plus it's always better having 2 programming languages opposed to 1. There is tons of support for XNA C#, e.g. XNA forums, MSDN esp this. Use the XNA Tutorials provided. It is a fantastic way of laying the foundations and making the C# language + XNA framework easy to understand.

If you are struggling with any problems, ask Phil. The man thinks in binary.

Start working on your game as soon as possible even if its gathering images/sound because it would help in the weeks to follow.

